1. **Stable Matching**

- stable matching problem
- Gale–Shapley algorithm
- hospital optimality
- context

Matching med-school students to hospitals

**Goal.** Given a set of preferences among hospitals and med-school students, design a self-reinforcing admissions process.

**Unstable pair.** Hospital $h$ and student $s$ form an unstable pair if both:
- $h$ prefers $s$ to one of its admitted students.
- $s$ prefers $h$ to assigned hospital.

**Stable assignment.** Assignment with no unstable pairs.
- Natural and desirable condition.
- Individual self-interest prevents any hospital–student side deal.

Stable matching problem: input

**Input.** A set of $n$ hospitals $H$ and a set of $n$ students $S$.
- Each hospital $h \in H$ ranks students.
- Each student $s \in S$ ranks hospitals.

<table>
<thead>
<tr>
<th>Hospitals' Preference Lists</th>
<th>Students' Preference Lists</th>
</tr>
</thead>
<tbody>
<tr>
<td>2nd: Yolanda 3rd: Zeus</td>
<td>2nd: Yolanda 3rd: Chicago</td>
</tr>
<tr>
<td>3rd: Zeus</td>
<td>3rd: Chicago</td>
</tr>
</tbody>
</table>

Xavier: favorite, Yolanda: least favorite

Atlanta, Boston, Chicago: hospitals

Xavier, Yolanda, Zeus: students

Atlanta, Xavier, Yolanda: hospitals’ preference lists

Xavier, Boston, Atlanta: students’ preference lists
Perfect matching

Def. A matching $M$ is a set of ordered pairs $h-s$ with $h \in H$ and $s \in S$ s.t.
- Each hospital $h \in H$ appears in at most one pair of $M$.
- Each student $s \in S$ appears in at most one pair of $M$.

Def. A matching $M$ is perfect if $|M| = |H| = |S| = n$.

Unstable pair

Def. Given a perfect matching $M$, hospital $h$ and student $s$ form an unstable pair if both:
- $h$ prefers $s$ to matched student.
- $s$ prefers $h$ to matched hospital.

Key point. An unstable pair $h-s$ could each improve by joint action.

Stable matching problem

Def. A stable matching is a perfect matching with no unstable pairs.

Stable matching problem. Given the preference lists of $n$ hospitals and $n$ students, find a stable matching (if one exists).
Stable roommate problem

Q. Do stable matchings always exist?
A. Not obvious a priori.

Stable roommate problem.
- 2n people; each person ranks others from 1 to 2n - 1.
- Assign roommate pairs so that no unstable pairs.

<table>
<thead>
<tr>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>B</td>
<td>C</td>
<td>A</td>
</tr>
<tr>
<td>C</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>D</td>
<td>A</td>
<td>B</td>
</tr>
</tbody>
</table>

Observation. Stable matchings need not exist.

Gale–Shapley deferred acceptance algorithm

An intuitive method that guarantees to find a stable matching.

**Gale–Shapley** (preference lists for hospitals and students)

**INITIALIZE** M to empty matching.

**WHILE** (some hospital h is unmatched and hasn’t proposed to every student)
  s ← first student on h’s list to whom h has not yet proposed.
  IF (s is unmatched)
    Add h–s to matching M.
  ELSE IF (s prefers h to current partner h’)
    Replace h’–s with h–s in matching M.
  ELSE
    s rejects h.

**RETURN** stable matching M.

Proof of correctness: termination

Observation 1. Hospitals propose to students in decreasing order of preference.

Observation 2. Once a student is matched, the student never becomes unmatched; only “trades up.”

Claim. Algorithm terminates after at most n^2 iterations of while loop.

Pf. Each time through the while loop, a hospital proposes to a new student. Thus, there are at most n^2 possible proposals.

<table>
<thead>
<tr>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
<th>5th</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
<th>5th</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>V</td>
<td>W</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td>V</td>
<td>E</td>
<td>C</td>
<td>D</td>
</tr>
<tr>
<td>B</td>
<td>W</td>
<td>X</td>
<td>Y</td>
<td>V</td>
<td>Z</td>
<td>W</td>
<td>C</td>
<td>D</td>
<td>E</td>
</tr>
<tr>
<td>C</td>
<td>X</td>
<td>Y</td>
<td>V</td>
<td>W</td>
<td>Z</td>
<td>X</td>
<td>D</td>
<td>E</td>
<td>A</td>
</tr>
<tr>
<td>D</td>
<td>Y</td>
<td>V</td>
<td>W</td>
<td>X</td>
<td>Z</td>
<td>Y</td>
<td>E</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>E</td>
<td>V</td>
<td>W</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td>Z</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
</tbody>
</table>

n(n-1) + 1 proposals
Proof of correctness: perfect matching

**Claim.** Gale–Shapley outputs a matching.

**Pf.**
- Hospital proposes only if unmatched. ⇒ matched to ≤ 1 student
- Student keeps only best hospital. ⇒ matched to ≤ 1 hospital

**Claim.** In Gale–Shapley matching, all hospitals get matched.

**Pf.** [by contradiction]
- Suppose, for sake of contradiction, that some hospital \( h \in H \) is unmatched upon termination of Gale–Shapley algorithm.
- Then some student, say \( s \in S \), is unmatched upon termination.
- By Observation 2, \( s \) was never proposed to.
- But, \( h \) proposes to every student, since \( h \) ends up unmatched. ※

**Claim.** In Gale–Shapley matching, all students get matched.

**Pf.** [by counting]
- By previous claim, all \( n \) hospitals get matched.
- Thus, all \( n \) students get matched. •

---

**Summary**

**Stable matching problem.** Given \( n \) hospitals and \( n \) students, and their preference lists, find a stable matching if one exists.


---

**Stable matching: quiz 2**

Do all executions of Gale–Shapley lead to the same stable matching?

A. No, because the algorithm is nondeterministic.
B. No, because an instance can have several stable matchings.
C. Yes, because each instance has a unique stable matching.
D. Yes, even though an instance can have several stable matchings and the algorithm is nondeterministic.
1. STABLE MATCHING

- stable matching problem
- Gale–Shapley algorithm
- hospital optimality
- context

Understanding the solution

For a given problem instance, there may be several stable matchings.

Understanding the solution

Def. Student \( s \) is a valid partner for hospital \( h \) if there exists any stable matching in which \( h \) and \( s \) are matched.

Ex.

- Both \( X \) and \( Y \) are valid partners for \( A \).
- Both \( X \) and \( Y \) are valid partners for \( B \).
- \( Z \) is the only valid partner for \( C \).

Stable matching: quiz 3

Who is the best valid partner for \( W \) in the following instance?

A. 
B. 
C. 
D. 

6 stable matchings

<table>
<thead>
<tr>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>B</td>
<td>Y</td>
<td>X</td>
</tr>
<tr>
<td>C</td>
<td>X</td>
<td>Y</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>W</td>
<td>Z</td>
</tr>
<tr>
<td>B</td>
<td>W</td>
<td>Y</td>
</tr>
<tr>
<td>C</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>D</td>
<td>Z</td>
<td>W</td>
</tr>
</tbody>
</table>

an instance with two stable matchings: \( S = \{ A-X, B-Y, C-Z \} \) and \( S' = \{ A-Y, B-X, C-Z \} \)
Def. Student $s$ is a valid partner for hospital $h$ if there exists any stable matching in which $h$ and $s$ are matched.

Hospital-optimal assignment. Each hospital receives best valid partner.

- Is it a perfect matching?
- Is it stable?

Claim. All executions of Gale–Shapley yield hospital-optimal assignment.

Corollary. Hospital-optimal assignment is a stable matching!

Student pessimal assignment. Each student receives worst valid partner.


Pf. [by contradiction]

- Suppose $h \rightarrow s$ matched in $M^*$ but $h$ is not the worst valid partner for $s$.
- There exists stable matching $M$ in which $s$ is paired with a hospital, say $h'$, whom $s$ prefers less than $h$.
  \[ \Rightarrow s \text{ prefers } h \text{ to } h'. \]
- Let $s'$ be the partner of $h$ in $M$.
- By hospital-optimality, $s$ is the best valid partner for $h$.
  \[ \Rightarrow h \text{ prefers } s \text{ to } s'. \]
- Thus, $h \rightarrow s$ is an unstable pair in $M$, a contradiction. •

Hospital optimality

Claim. Gale–Shapley matching $M^*$ is hospital-optimal.

Pf. [by contradiction]

- Suppose a hospital is matched with student other than best valid partner.
- Hospitals propose in decreasing order of preference.
  \[ \Rightarrow \text{some hospital is rejected by a valid partner during Gale–Shapley} \]
- Let $h$ be first such hospital, and let $s$ be the first valid partner that rejects $h$.
- Let $M$ be a stable matching where $h$ and $s$ are matched.
- When $s$ rejects $h$ in Gale–Shapley, $s$ forms (or re-affirms) commitment to a hospital, say $h'$.
  \[ \Rightarrow s \text{ prefers } h' \text{ to } h. \]
- Let $s'$ be partner of $h'$ in $M$.
- $h'$ had not been rejected by any valid partner (including $s'$) at the point when $h$ is rejected by $s$.
  \[ \Rightarrow h' \text{ prefers } s' \text{ to } s. \]
- Thus, $h' \rightarrow s'$ is unstable in $M$, a contradiction. •

Stable matching: quiz 4

Suppose each agent knows the preference lists of every other agent before the hospital propose-and-reject algorithm is executed.
Which is true?

- A. No hospital can improve by falsifying its preference list.
- B. No student can improve by falsifying their preference list.
- C. Both A and B.
- D. Neither A nor B.
1. Stable Matching

- Stable matching problem
- Gale–Shapley algorithm
- Hospital optimality
- Context

Extensions

Extension 1. Some agents declare others as unacceptable.
Extension 2. Some hospitals have more than one position.
Extension 3. Unequal number of positions and students.

Def. Matching \( M \) is unstable if there is a hospital \( h \) and student \( s \) such that:
- \( h \) and \( s \) are acceptable to each other; and
- Either \( s \) is unmatched, or \( s \) prefers \( h \) to assigned hospital; and
- Either \( h \) does not have all its places filled, or \( h \) prefers \( s \) to at least one of its assigned students.

Theorem. There exists a stable matching.


2012 Nobel Prize in Economics


Alvin Roth. Applied Gale–Shapley to matching med-school students with hospitals, students with schools, and organ donors with patients.
How Game Theory Helped Improve New York City’s High School Application Process

By TRACY TULLIS  DEC. 5, 2014

Tuesday was the deadline for eighth graders in New York City to submit applications to secure a spot at one of 456 public high schools. After months of school tours and tests, auditions and interviews, 75,000 students have entrusted their choices to a computer program that will arrange their school assignments for the coming year. The weeks of research and deliberation will be reduced to a fraction of a second of mathematical calculation: In just a couple of hours, all the sorting for the Class of 2019 will be finished.

A modern application

Content delivery networks. Distribute much of world’s content on web.

User. Preferences based on latency and packet loss.

Web server. Preferences based on costs of bandwidth and co-location.

Goal. Assign billions of users to servers, every 10 seconds.

Algorithmic Nuggets in Content Delivery

Bruce M. Maggs
Gale J. A. Krasnoff
larry@cs.duke.edu

Ramesh K. Sitaraman
Univ. Arizona and Akamai
ramesh@cs.arizona.edu

ABSTRACT

This paper ‘provably make the right’ in the solutions that provide the lowest bandwidth of a trading context. Algorithmic methods, based on our experiences in building one of the world’s largest content delivery networks, have been applied to formulate the local functions and online peers. These algorithms are then employed in the overlay routing network, and are chosen in certain scenarios. In such networks, we first employ the theory matching the local function to online peers. The requested object is served by the theoretical solutions, and finally, which is implemented in practice. Through these examples, we highlight the role of algorithmic research in the design of real, high-speed global networks to the ready of digital media. The research also presents new problems and practical requirements that have driven research.